M252 實作Google通訊協定緩衝區

NuMicro® 32 位系列微控制器範例程式碼介紹

文件資訊

應用簡述	本範例程式碼演示如何使用 Nanopb 在 M252 實作通訊協定緩衝區。Nanopb 是一個輕量、支持 C 語言的 Google 通訊協定緩衝區資料格式的實現。
BSP 版本	M251/M252/M254/M256/M258 Series CMSIS BSP V3.01.002
開發平台	NuMaker-M252SD Ver1.2

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller and microprocessor based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

1 概述

通訊協定緩衝區 Protocol Buffers（簡稱：ProtoBuf）是一種序列化資料結構的協定，由 Google 所開發。對於透過管道(Pipeline)或儲存資料進行通訊的程式開發上是很有用的。這個方法包含一個介面描述語言，描述一些資料結構，並提供程式工具根據這些描述產生程式碼，以產生資料結構或解析資料串流。

微控制器 M252 提供範例程式碼，示範如何實作出通訊協定緩衝區功能。請注意，此範例程式也適用於M251/M254/M256/M258 系列。

範例程式碼使用 Petteri Aimonen 的 Nanopb 開源函式庫。Nanopb下載地址 https://jpa.kapsi.fi/nanopb/download/ 。

1.1 開源程式碼 Nanopb 使用

Nanopb 是一個 ANSI-C 函式庫，用於以 Google 的通訊協定緩衝區格式對消息進行編碼和解碼，對記憶體和程式碼空間的要求最低。它主要適用於 32 位微控制器，但也適用於其他具有嚴格（<10 kB ROM，<1 kB RAM）記憶體限制的嵌入式系統。

我們這裡使用nanopb-0.4.5-windows-x86。首先，使用 Sample Template 創建一個 Protobuf 專案，然後把nanopb-0.4.5-windows-x86 目錄下七個檔案 pb.h, pb_common.c, pb_common.h, pb_decode.c, pb_decode.h, pb_encode.c, pb_encode.h 複製到專案資料夾中，如圖1-1 專案資料夾內容所示：

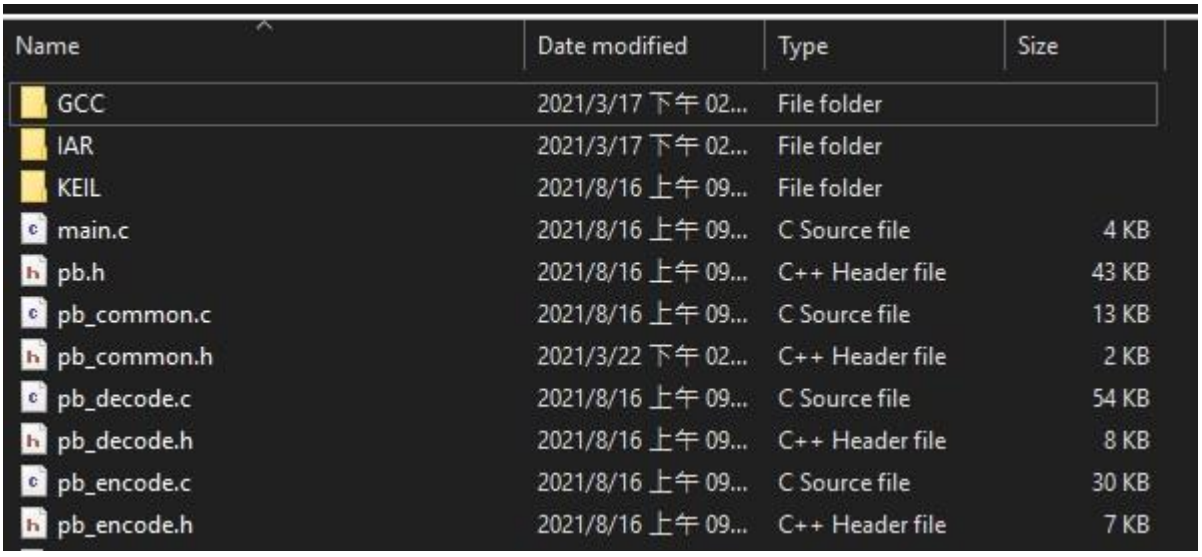

圖 1-1 專案資料夾內容

Protobuf 的核心是一個 .proto 文件，我們自定義一個 .proto 來創建我們的協議資料，然後使用 protoc 工具編譯產生 C 程式碼，有兩個文件：一個標頭檔、一個原始檔。 其中 protoc 工具在我們下載的 nanopb-0.4.5-windows-x86 的 generator-bin 資料夾中，如圖1-2 protoc 工具在 generator-bin 資料夾 所示：

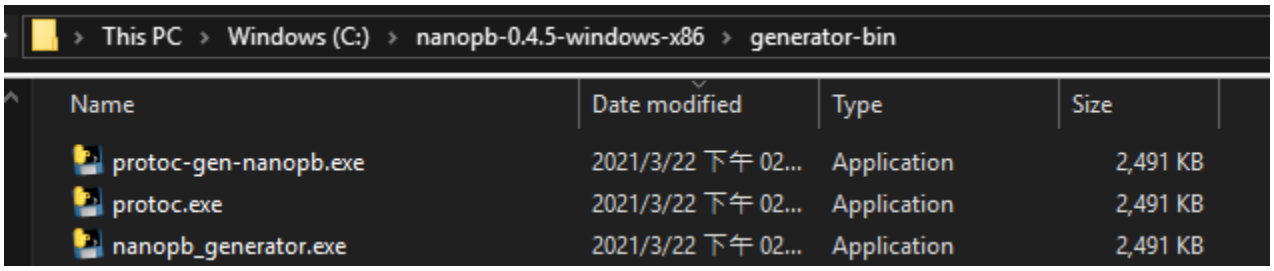

圖 1-2 protoc 工具在 generator-bin 資料夾

為了能在命令列中任何路徑下使用 protoc 工具，我們需要把 protoc.exe 所在路徑添加到環境變數中。然後，我們在 Protobuf 資料夾中創建一個 test.proto 檔案。

```
syntax = "proto2";

message Student
{
 required uint32 num      = 1;
 required uint32 py_score = 2;
 required uint32 c_score  = 3;
}
```

進入命令列視窗，在命令列 cd 進入 test.proto 所在目錄；輸入如下命令編譯 test.proto： protoc --nanopb_out=. test.proto 產生 test.pb.c 與 test.pb.h 檔案，如圖1- 3 命令列視窗 與 圖1-4 專案資料夾內容 所示：

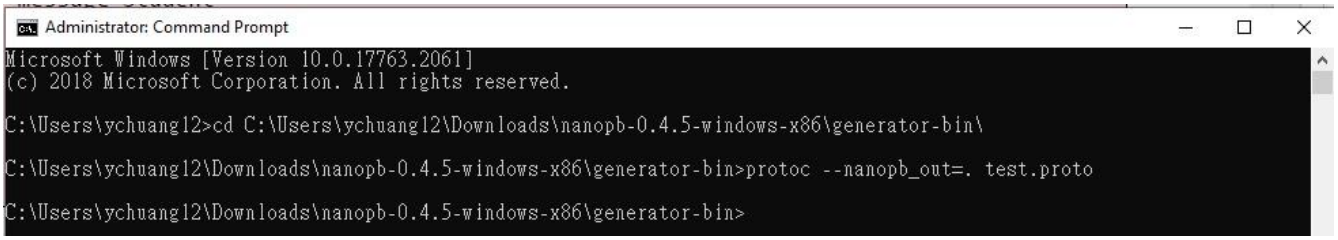

圖 1- 3 命令列視窗

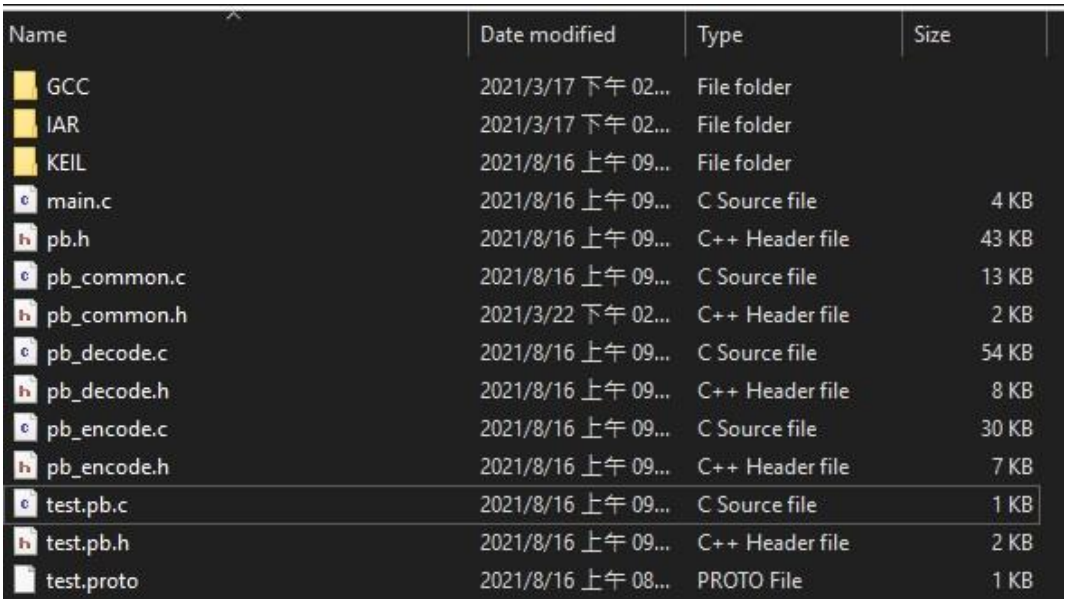

圖 1-4 專案資料夾內容

增加標頭檔搜索路徑到專案，並把 Protobuf 資料夾裡的檔案添加到 mdk 環境，如圖1-5 編譯環境所示：

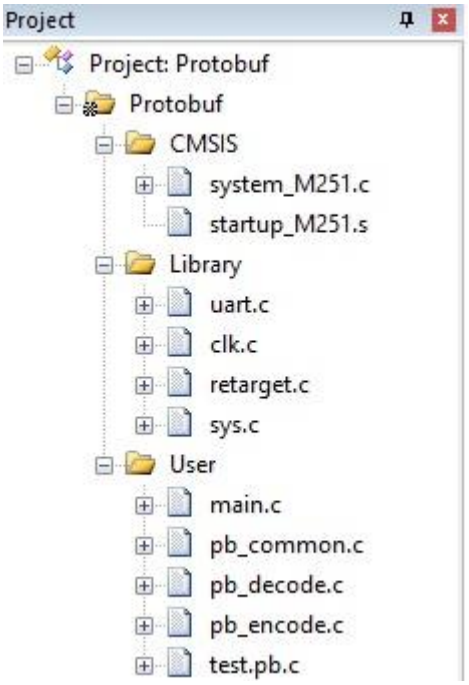

圖 1-5 編譯環境

Demo 很簡單，編碼就是建構一個協議資料結構，調用編碼往 buffer 中寫入資料；解碼正好反過來，從 buffer 中讀出資料解碼到結構裡。其中，編碼相關的函式在 pb_encode.c 中，解碼相關的函式在 pb_decode.c 中。

1.2 執行結果

執行結果，以及列印資訊如圖1-6 執行結果，以及列印資訊所示。

圖1-6 執行結果，以及列印資訊

2 程式碼介紹

一開始初始化 pack_stu，編碼後透過 o_stream 進行傳送，再經由 i_stream 接收進行解碼，得出資料寫入 unpack_stu，比對結果如圖 1-6 執行結果，以及列印資訊：

```
        /* pack_stu and unpack_stu is Student type*/
        Student pack_stu = {0};
        Student unpack_stu = {0};
        /* pack_stu initialization */
        pack_stu.num  = 1024;
        pack_stu.py_score = 4096;
        pack_stu.c_score = 512;
        /* o_stream initialization */
        o_stream = pb_ostream_from_buffer(buffer, sizeof(buffer));
        /* encode all pack_stu fields from Student into o_stream */
        pb_encode(&o_stream, Student_fields, &pack_stu);
        /* i_stream initialization */
        i_stream = pb_istream_from_buffer(buffer, sizeof(buffer));
        /* decode all fields from i_stream into unpack_stu */
        pb_decode(&i_stream, Student_fields, &unpack_stu);
        /* print unpack_stu result */
        printf("unpack_stu.num = %d\n", unpack_stu.num);
        printf("unpack_stu.py_score = %d\n", unpack_stu.py_score);
        printf("unpack_stu.c_score = %d\n", unpack_stu.c_score);
```

編碼所有欄位：

```
bool checkreturn pb_encode(pb_ostream_t *stream, const pb_msgdesc_t *fields, const void
*src_struct)
{
    pb_field_iter_t iter;

    if (!pb_field_iter_begin_const(&iter, fields, src_struct))
        return true; /* Empty message type */

    do
    {
        if (PB_LTYPE(iter.type) == PB_LTYPE_EXTENSION)
        {
            /* Special case for the extension field placeholder */
            if (!encode_extension_field(stream, &iter))
                return false;
```

```
        }
        else
        {
            /* Regular field */
            if (!encode_field(stream, &iter))
                return false;
        }
    } while (pb_field_iter_next(&iter));

    return true;
}
```

解碼所有欄位：

```
bool checkreturn pb_decode(pb_istream_t *stream, const pb_msgdesc_t *fields, void
*dest_struct)
{
    bool status;

    status = pb_decode_inner(stream, fields, dest_struct, 0);

#ifdef PB_ENABLE_MALLOC
    if (!status)
        pb_release(fields, dest_struct);
#endif

    return status;
}
```

3 軟體與硬體需求

3.1 軟體需求

- BSP 版本
  - M251/M252/M254/M256/M258 Series CMSIS BSP V3.01.002
- IDE 版本
  - Keil uVersion 5.26

3.2 硬體需求

- 電路元件
  - NuMaker-M252SD Ver1.2
- 示意圖

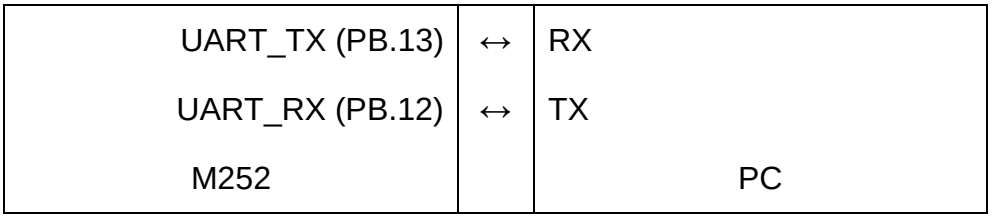

圖 3-1 線路連接

4 目錄資訊

EC_M252_Protocol_Buffers_V1.00	
Library	Sample code header and source files
CMSIS	Cortex® Microcontroller Software Interface Standard (CMSIS) by Arm® Corp.
Device	CMSIS compliant device header file
StdDriver	All peripheral driver header and source files
SampleCode	
ExampleCode	Source file of example code

圖 4-1 目錄資訊

5 範例程式執行

1. 根據目錄資訊章節進入 ExampleCode 路徑中的 KEIL 資料夾，雙擊 M252_Protocol_Buffers.uvprojx。
2. 進入編譯模式介面
   a. 編譯
   b. 下載代碼至記憶體
   c. 進入 / 離開除錯模式
3. 進入除錯模式介面
   a. 執行代碼

6 修訂紀錄

Date	Revision	Description
2021.09.24	1.00	1. 初始發布.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.